

In [35]:

```
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from pylab import *
```

tetta = 5

In [101]:

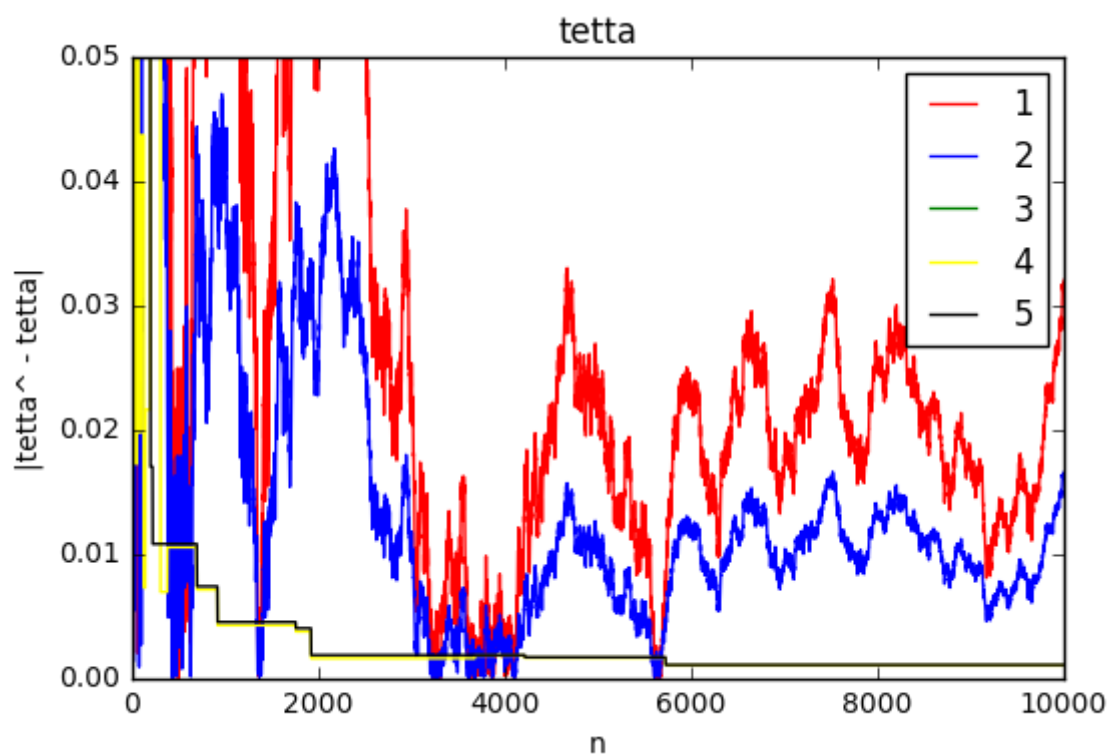
```
N = 10000
tetta = 5
# Выборка из равномерного распределения с параметром tetta
s = np.random.uniform(0,tetta,N)

# Массив для первой оценки
arr_1 = []
# Массив для второй оценки
arr_2 = []
# Массив для третьей оценки
arr_3 = []
# Массив для четвертой оценки
arr_4 = []
# Массив для пятой оценки
arr_5 = []

# Подсчет разностей для разных оценок в зависимости от n
for i in range(1, N + 1):
    arr_1 += [abs(s[:i].sum() / i * 2.0 - tetta)]
    arr_2 += [abs(s[:i].sum() / i + s[:i].max() / 2.0 - tetta)]
    arr_3 += [abs((i + 1) * s[:i].min() * 1.0 - tetta)]
    arr_4 += [abs(s[:i].max() + s[:i].min() - tetta)]
    arr_5 += [abs((i + 1) / i * 1.0 * s[:i].max() - tetta)]

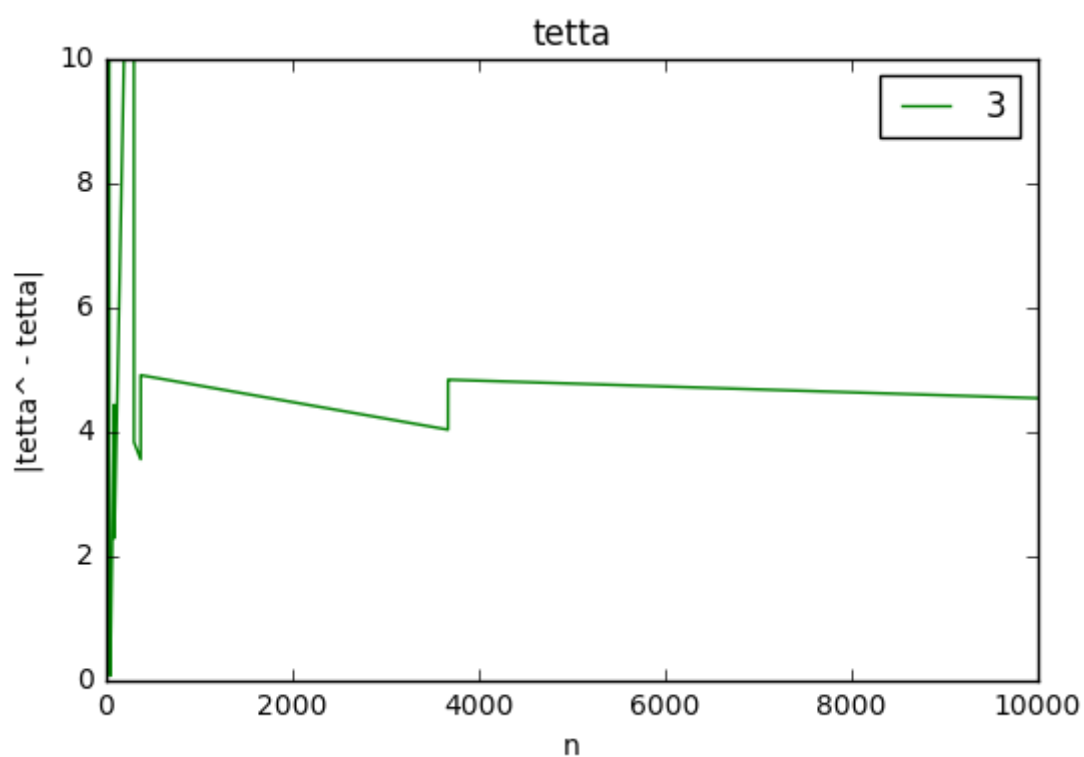
# Построение графиков
ylim(0,0.05)
plt.plot(range(1, N + 1), arr_1, color = 'red', label='1')
plt.plot(range(1, N + 1), arr_2, color = 'blue', label='2')
plt.plot(range(1, N + 1), arr_3, color = 'green', label='3')
plt.plot(range(1, N + 1), arr_4, color = 'yellow', label='4')
plt.plot(range(1, N + 1), arr_5, color = 'black', label='5')

plt.legend(loc='best')
plt.title('tetta')
plt.xlabel('n')
plt.ylabel('|tetta^ - tetta|');
plt.show()
```



In [102]:

```
# График для 3й оценки
ylim(0,10)
plt.plot(range(1, N + 1), arr_3, color = 'green', label='3')
plt.legend(loc='best')
plt.title('tetta')
plt.xlabel('n')
plt.ylabel('|tetta^ - tetta|');
plt.show()
```



tetta = 50

In [104]:

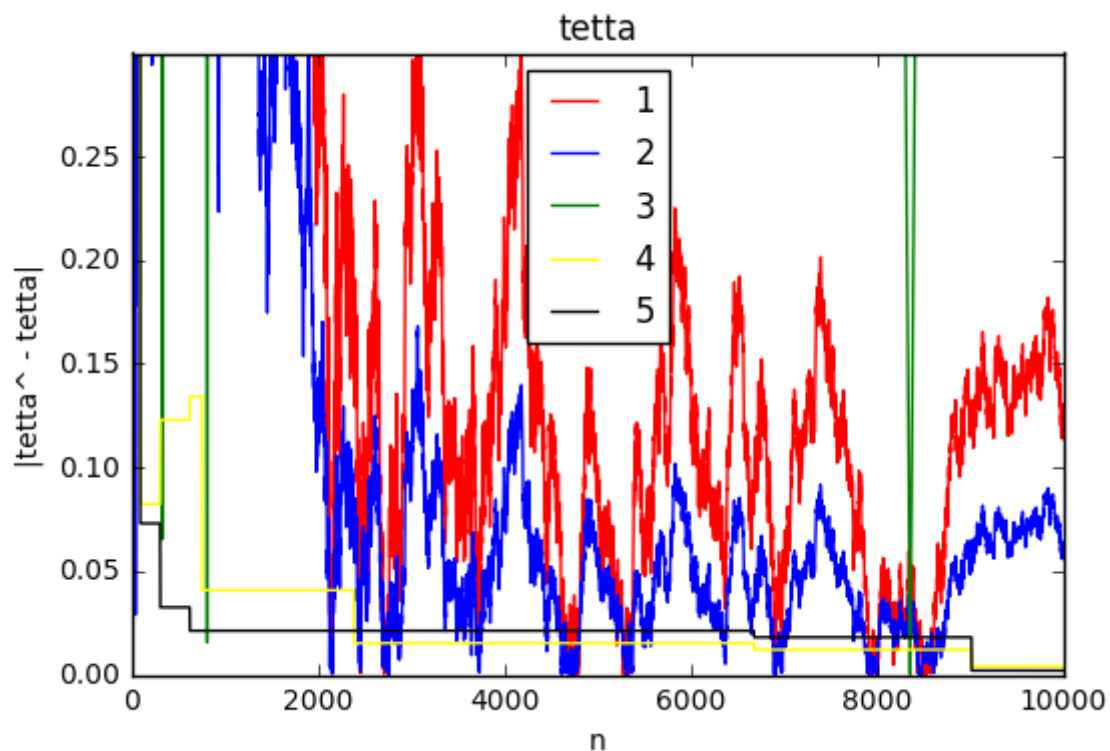
```
N = 10000
tetta = 50
# Выборка из равномерного распределения с параметром tetta
s = np.random.uniform(0,tetta,N)

# Массив для первой оценки
arr_1 = []
# Массив для второй оценки
arr_2 = []
# Массив для третьей оценки
arr_3 = []
# Массив для четвертой оценки
arr_4 = []
# Массив для пятой оценки
arr_5 = []

# Подсчет разностей для разных оценок в зависимости от n
for i in range(1, N + 1):
    arr_1 += [abs(s[:i].sum() / i * 2.0 - tetta)]
    arr_2 += [abs(s[:i].sum() / i + s[:i].max() / 2.0 - tetta)]
    arr_3 += [abs((i + 1) * s[:i].min() * 1.0 - tetta)]
    arr_4 += [abs(s[:i].max() + s[:i].min() - tetta)]
    arr_5 += [abs((i + 1) / i * 1.0 * s[:i].max() - tetta)]

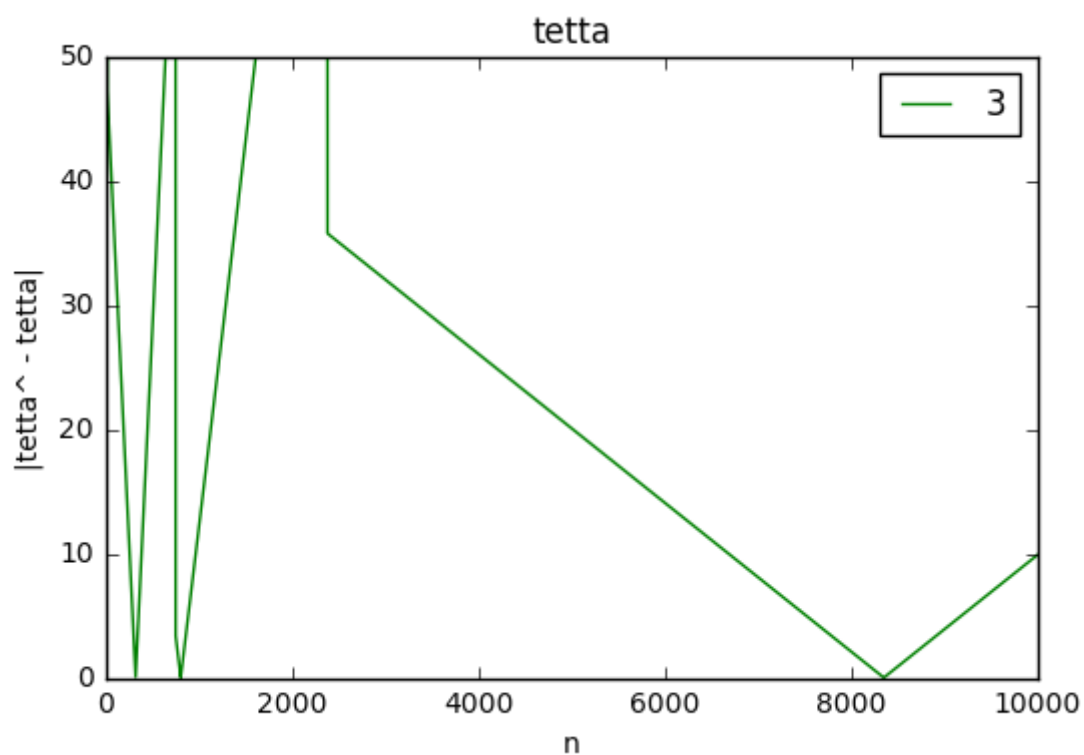
# Построение графиков
ylim(0,0.3)
plt.plot(range(1, N + 1), arr_1, color = 'red', label='1')
plt.plot(range(1, N + 1), arr_2, color = 'blue', label='2')
plt.plot(range(1, N + 1), arr_3, color = 'green', label='3')
plt.plot(range(1, N + 1), arr_4, color = 'yellow', label='4')
plt.plot(range(1, N + 1), arr_5, color = 'black', label='5')

plt.legend(loc='best')
plt.title('tetta')
plt.xlabel('n')
plt.ylabel('|tetta^ - tetta|');
plt.show()
```



In [107]:

```
# График для 3й оценки
ylim(0,50)
plt.plot(range(1, N + 1), arr_3, color = 'green', label='3')
plt.legend(loc='best')
plt.title('tetta')
plt.xlabel('n')
plt.ylabel('|tetta^ - tetta|');
plt.show()
```



tetta = 200

In [109]:

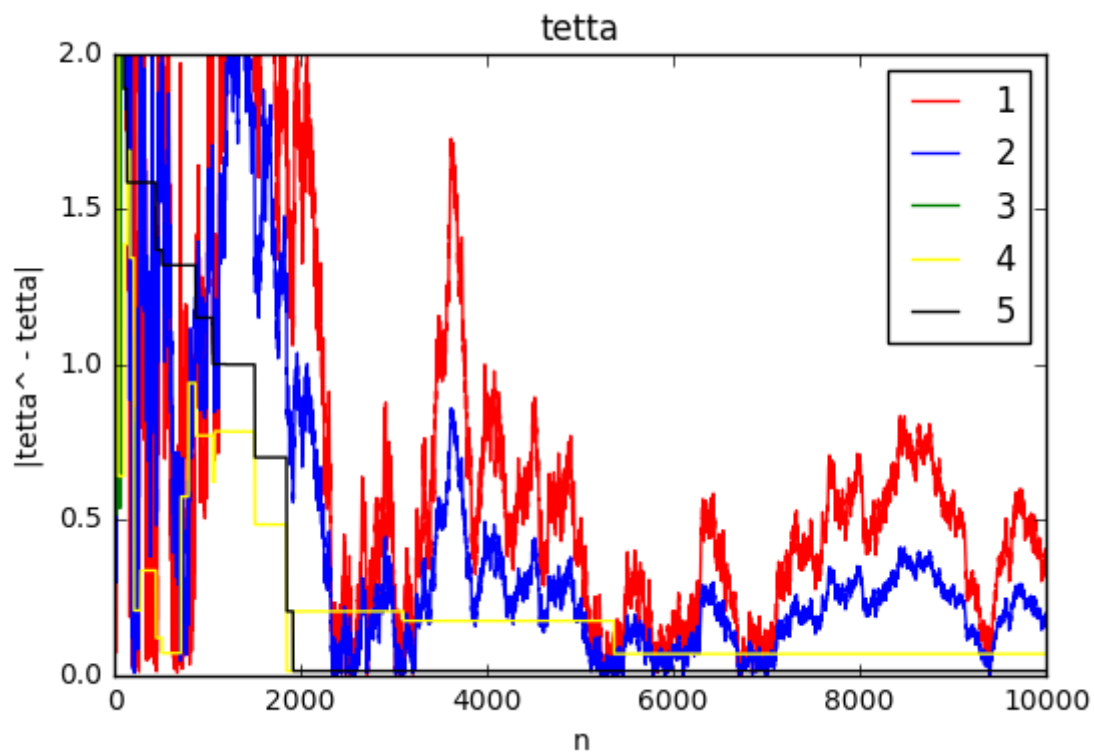
```
N = 10000
tetta = 200
# Выборка из равномерного распределения с параметром tetta
s = np.random.uniform(0,tetta,N)

# Массив для первой оценки
arr_1 = []
# Массив для второй оценки
arr_2 = []
# Массив для третьей оценки
arr_3 = []
# Массив для четвертой оценки
arr_4 = []
# Массив для пятой оценки
arr_5 = []

# Подсчет разностей для разных оценок в зависимости от n
for i in range(1, N + 1):
    arr_1 += [abs(s[:i].sum() / i * 2.0 - tetta)]
    arr_2 += [abs(s[:i].sum() / i + s[:i].max() / 2.0 - tetta)]
    arr_3 += [abs((i + 1) * s[:i].min() * 1.0 - tetta)]
    arr_4 += [abs(s[:i].max() + s[:i].min() - tetta)]
    arr_5 += [abs((i + 1) / i * 1.0 * s[:i].max() - tetta)]

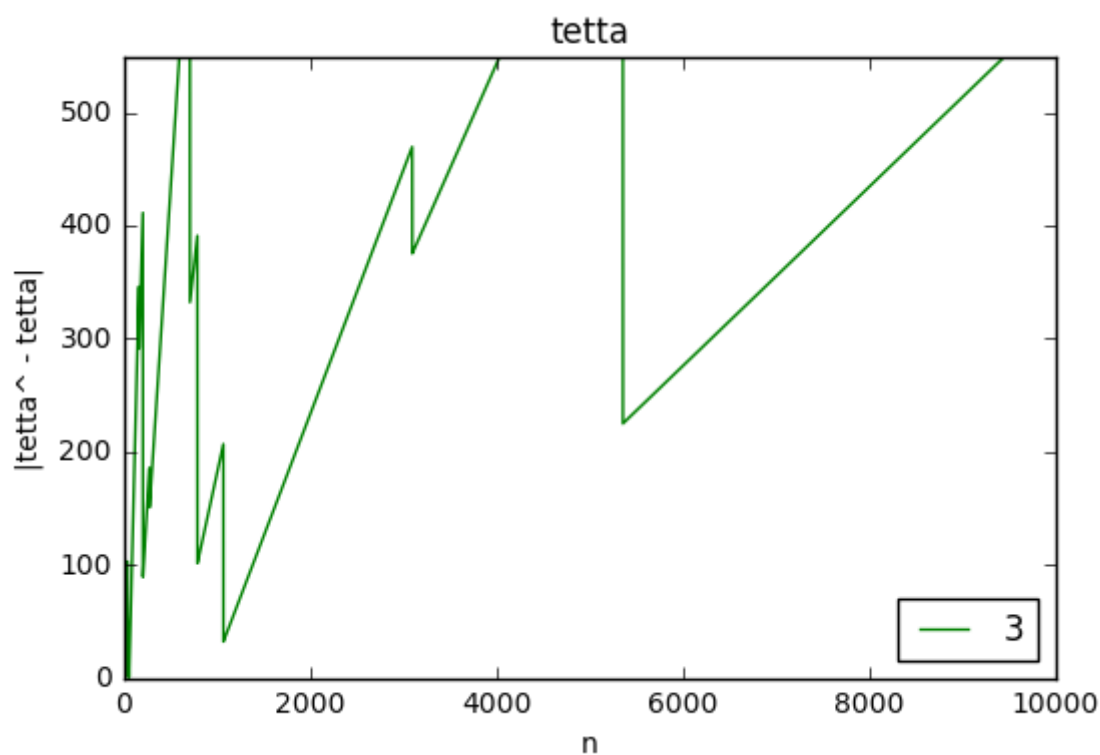
# Построение графиков
ylim(0,2)
plt.plot(range(1, N + 1), arr_1, color = 'red', label='1')
plt.plot(range(1, N + 1), arr_2, color = 'blue', label='2')
plt.plot(range(1, N + 1), arr_3, color = 'green', label='3')
plt.plot(range(1, N + 1), arr_4, color = 'yellow', label='4')
plt.plot(range(1, N + 1), arr_5, color = 'black', label='5')

plt.legend(loc='best')
plt.title('tetta')
plt.xlabel('n')
plt.ylabel('|tetta^ - tetta|');
plt.show()
```

In [112]:

```
# График для 3й оценки
ylim(0,550)
plt.plot(range(1, N + 1), arr_3, color = 'green', label='3')
plt.legend(loc='best')
plt.title('tetta')
plt.xlabel('n')
plt.ylabel('|tetta^ - tetta|');
plt.show()
```



Вывод

В плане модули разности оценки и истинного значения параметра лучше всего себя показали 4 и 5 оценки